Power System Cascading Failure Mitigation by Reinforcement Learning

Yongli Zhu
Texas A&M University, USA
June.27.2021
Outline

• 1. Motivation of Multi-Stage Cascading Failure
• 2. Formulation of Multi-Stage Cascading Failure
• 3. Mitigation Strategy by RL
• 4. Case Study
• 5. Conclusions and Future Work
1. Motivation of Multi-stage Cascading Failure

- Single-Stage Cascading Failure problem has been widely studied by power systems community

- However, succeeding outage stages can happen one by one closely, e.g. a wind storm happens first, then followed by the mis-operation of human operators ➔ Thus, Multi-Stage Cascading Failure (MSCF) problem is proposed.
1. Motivation of Multi-stage Cascading Failure

- Can we use any control strategy to mitigate (limit or reduce) such kind of cascading failures? => Yes
  - Strategy options: load shedding, generation adjustment, line switching, transformer tap-ratio change, etc.
- How to determine which control strategy to use and when to use?
  - 1) Conventional approach like SCOPF may be useful for Single-Stage Cascading Failure problem
  - 2) However, for Multi-Stage Cascading Failure, both the timing (order) and type of the consecutive attacks (e.g. faults) can be unknown or stochastic. Only using SCOPF may not handle the MSCF problem well.
- We can resort to data-driven / machine learning methods
- Inspiration from Alpha-Go by Google
2. Formulation of Multi-Stage Cascading Failure

- **Generation**: one “event” of the cascading failures within one stage, e.g. a line tripping.

- **Stage**: after an attack (e.g. one line is broken by a natural disaster), the grid evolves with a series of potential *generations*. Finally, the power system will either reach a new equilibrium point if it exists; or the system collapses.

Example simulation results of the IEEE 118-bus (= node) system for a two-stage MSCF problem in two independent episodes:

* ACPF (alternative current power flow): a set of nonlinear equations that a power grid needs to satisfy when it reaches steady state.

---

**Table 1. Result of Episode-1**

<table>
<thead>
<tr>
<th>Stage-1</th>
<th>ACPF converge</th>
<th>Over limit Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation-1</td>
<td>Yes</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 2. Result of Episode-2**

<table>
<thead>
<tr>
<th>Stage-1</th>
<th>ACPF converge</th>
<th>Over limit Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation-1</td>
<td>Yes</td>
<td>2</td>
</tr>
<tr>
<td>Generation-2</td>
<td>Yes</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage-2</th>
<th>ACPF converge</th>
<th>Over limit Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generation-1</td>
<td>Yes</td>
<td>4</td>
</tr>
<tr>
<td>Generation-2</td>
<td>Yes</td>
<td>2</td>
</tr>
<tr>
<td>Generation-3</td>
<td>Yes</td>
<td>2</td>
</tr>
<tr>
<td>Generation-4</td>
<td>Yes</td>
<td>3</td>
</tr>
<tr>
<td>Generation-5</td>
<td>Yes</td>
<td>10</td>
</tr>
<tr>
<td>Generation-6</td>
<td>Yes</td>
<td>20</td>
</tr>
<tr>
<td>Generation-7</td>
<td>No</td>
<td>--</td>
</tr>
</tbody>
</table>

| Result | Win |

---

\[ 0 = -P_i + \sum_{k=1}^{N} |V_i||V_k|(G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) \]

\[ 0 = -Q_i + \sum_{k=1}^{N} |V_i||V_k|(G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \]

* https://en.wikipedia.org/wiki/Power-flow_study
2. Formulation of Multi-stage Cascading Failure

- Mimicking the corrective controls by DCOPF

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in G} c_i p_i + \sum_{j \in D} d_j (p_j - P_{dj}) \\
\text{subject to} & \quad F = Ap \\
& \quad \sum_{k=1}^{n} p_k = 0 \\
& \quad P_{dj} \leq p_j \leq 0, \quad j \in D \\
& \quad P_{gi}^{\min} \leq p_j \leq P_{gi}^{\max}, \quad i \in G \\
& \quad -F_{L}^{\max} \leq F_l \leq F_{L}^{\max}, \quad l \in L
\end{align*}
\]

- \( c_i, d_j \): generation cost / load shedding cost per unit power (e.g., $/MW); \( p_i \): generator power (MW)
- \( P_{dj} \): original load power (MW); \( p_j \): load power (MW) (here the sign of electric power is negative for load)
- \( A \): a constant matrix to associate the net nodal power injections with the branch power flows.
- \( F \): a vector of all the branch flows; \( p = [p_k], k = 1 \ldots n \): represents the net nodal power injections.
- \( n \): the total bus number; \( G, D, L \): respectively the generator set, load set and branch set
3. Mitigation Strategy by RL

Applying RL/DRL in Cascading Failure Mitigation

1) **Reward design** (of each Stage)
   - Total generation cost (i.e. the negative objective function value of DCOPF) (if converge);
   - −1000, if DCOPF or ACPF diverge;
   - +1000, if system finally reaches a new steady state at the last stage.

2) **Action design**
   - In the previous DCOPF formulation, the “branch flow limit” $F_{L_{max}}$ is adopted as the action.

3) **State design**
   - $[\text{branch\_loading\_status}, V_1, \theta_1, P_1, Q_1, \ldots, V_n, \theta_n, P_n, Q_n]$ (voltage magnitude, voltage angle, active power, reactive power)

Figure 1. The overall workflow of grid simulation for MSCF study.
4. Case Study

- **Test power grid:**
  - IEEE 118-bus system

- **Network-1:** SARSA (On-policy TD)
  - **Shallow Neural Network (RL)**

- **Network-2:** Q-learning (Off-policy TD)
  - **Deep Neural Network (DRL)**

![Network architecture diagram]

It contains:
- 137 buses (nodes)
  - 19 generators buses (red dots)
  - 91 loads buses
- 186 lines (parallel lines included)

Network architecture is:
- one input layer, one output player
- one hidden layer with 10 neuron units

Input:
- a 1-D vector with 753 (=137×4+177+28) elements

Output:
- the action in the RL framework (i.e., the line flow limit $F_{L_{max}}$)

*Action* is bounded by [0.80, 1.25]

- **Image-like input:** $784 = 28 \times 28$ (extend the original input (length = 753) by padding extra zeros)

The output of the 2nd-last layer (dim 1×10) is used in both $\epsilon$-greedy and greedy policies

The candidate set of *Action*:
- [0.8, 0.85, 0.9, 0.95, 1.0, 1.05, 1.1, 1.15, 1.20, 1.25]

![Network structure diagram]

Figure 3. The network structure used in Deep RL.

Figure 4. The shallow neural network structure used in RL.
4. Case Study

Table 3. Learning Performance

<table>
<thead>
<tr>
<th>PERFORMANCE</th>
<th>SHALLOW NETWORK</th>
<th>DEEP NETWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win rate</td>
<td>78.00%</td>
<td>78.07%</td>
</tr>
<tr>
<td>Avg. reward</td>
<td>640.08</td>
<td>630.46</td>
</tr>
</tbody>
</table>

Maximum episode number = 10000 (for both networks)
Learning rate = 0.0001, and the discount rate $\gamma = 0.7$
Maximum stage number = 3

It can be observed that:

1) Both RL and Deep RL have achieved satisfactory results in terms of winning rates (i.e., *fewer system collapses*).

2) The higher the average winning rate, the lower the average reward may become; and vice versa.
   * One explanation is: if the system operator (RL agent) is willing to shed (cut) more load then the system typically recovers faster (i.e. toward winning); but that way will also increase the obj. function (thus reduce the average reward).
5. Conclusions and Future Work

• A Multi-Stage Cascading Failure (MSCF) problem is proposed and formulated

• A systematic (deep) RL framework is designed for the mitigation of MSCF problem.

• The proposed RL-based mitigation strategy works effectively on the IEEE 118-bus system under both shallow and deep architectures.

• Future work
  • Investigate effects of hyper-parameters (layer numbers, learning rate, discount factor, etc.) of the neural networks on the mitigation performance
  • Consider more control options e.g. transformer tap ratio, energy storage, etc.